Claims

1. A compiler for converting source code for a program written in a programming language into an object program written in a machine language comprising:

an optimization execution unit for performing optimization of said object program written in said machine language; and

a program modification unit for, before said optimization process is performed by said optimization execution unit, modifying said object program to provide a form that is appropriate for said optimization,

wherein, when said object program includes a branch, said program modification unit selects at said branch a specific path to extract, relative to said branch, a series of paths that are not merged.

2. The compiler according to claim 1, wherein said optimization execution unit performs said optimization for said paths that are extracted by said program modification unit.

3. The compiler according to claim 1, wherein, when differences in execution frequencies depend on a plurality of paths at said branch in said object program, said

program modification unit selects a path having a higher execution frequency, and

relative to said branch, extracts a series of paths that are not merged.

4. A compiler, for converting source code for a program written in a

programming language into an object program in a machine language, comprising:

5   a path determiner, for selecting, when said object program in said machine

language includes a branch, a specific path at said branch for determining a series of

target paths for optimization;

a control flow graph modification unit, for modifying a control flow graph for

said object program to separate said optimized target paths from other paths; and

10   an optimization execution unit for employing said control flow graph obtained by

said control flow graph modification unit to perform optimization of said optimization

target paths.

5. The compiler according to claim 4, wherein said control flow graph

modification unit extracts said optimization target paths from said control flow graph for

15   said object program, and defines said optimization target paths as paths that, relative to

said branch, are not to be merged; and copies path segments that are included in said

optimization target paths and connects said copies of said path segments to paths other

than those for said optimization target paths, thereby separating said control flow graph into said optimization target paths and into paths that are formed using said copies of said path segments and said paths that are not included in said optimization target paths.

6. The compiler according to claim 5, wherein, from among the paths included in said optimization target paths, said control flow graph modification unit copies path segments at locations other than those whereat said object program branches into a plurality of paths, and connects the copies of said path segments to said paths that are not included in said optimization target paths.

7. A computer system that includes a compiler, for converting source code for a program written in a programming language into an object program in a machine language, comprising:

a path determiner, for selecting, when said object program in said machine language includes a branch, a specific path at said branch for determining a series of target paths for optimization;

a control flow graph modification unit, for modifying a control flow graph for said object program to separate said optimized target paths from other paths; and

an optimization execution unit for employing said control flow graph obtained by said control flow graph modification unit to perform optimization of said optimization target paths.

8. An optimization method, for performing optimization to improve process efficiency of a program, comprising the steps of:

selecting, when a program includes a branch, a specific path at said branch for determining a series of target paths for optimization;

modifying a control flow graph for said program to separate said optimized target paths from other sub-paths; and

employing said control flow graph obtained by said control flow graph modification unit to perform optimization of said optimization target paths.

9. The optimization method according to claim 8, wherein said step of modifying said control flow graph includes steps of:

copying all optimization target paths, for a sub-path that is led to an optimization target path, that can be reached from a point whereto said sub-path is led;

copying all the paths that extend from said point whereto said sub-path is led to the copied paths;

changing the connections of all the edges of said optimization target path from

said sub-path to the connections for said copied paths;

changing, when there is an edge that externally flows to said optimization target

path and when said edge is said edge of said sub-path, the connection of said edge

5   connecting said edge to said copied paths; and

forming a merging point, consisting of a first block at a starting point for said

edge and a second block along a copied path that corresponds to said first block, and

replacing an edge that flows out from said optimization target path with an edge from

said merging point.

10   10. The optimization method according to claim 9, wherein, at said two steps of

copying paths, the copying of a path is eliminated for a location whereat said program

branches into a plurality of paths.

11. An optimization method for performing optimization in order to improve the

processing efficiency of a program comprising the steps of:

15   when a program includes a branch, selecting at said branch a specific path from

which to extract, relative to said branch, a series of paths that are not merged; and

performing optimization for said paths that are selected.

12. The optimization method according to claim 11, wherein, at said step of extracting paths from said program, when differences in execution frequencies depend on a plurality of paths at said branch in said object program, a path having a higher execution frequency is selected, and a series of paths that are not merged are selected relative to said branch.

13. An optimization program for permitting a computer to perform optimization of an object program, which comprises:

a process for selecting, when said object program in said machine language includes a branch, a specific path at said branch for determining a series of target paths for optimization;

a process for modifying a control flow graph for said object program to separate said optimized target paths from other sub-paths; and

a process for employing said obtained control flow graph to perform optimization of said optimization target paths.

14. A storage medium on which input means for a computer stores a computer-readable program that permits said computer to perform:

a process for selecting, when said object program in said machine language includes a branch, a specific path at said branch for determining a series of target paths for optimization;

a process for modifying a control flow graph for said object program to separate

5 said optimized target paths from other sub-paths; and

a process for employing said obtained control flow graph to perform optimization of said optimization target paths.

15. A program transmission apparatus comprising:

storage means for storing a program that permits a computer to perform

10 a process for selecting, when said object program in said machine language includes a branch, a specific path at said branch for determining a series of target paths for optimization,

a process for modifying a control flow graph for said object program to separate said optimized target paths from other sub-paths, and

15 a process for employing said obtained control flow graph to perform optimization of said optimization target paths; and

transmission means for the reading of said program from said storage means and the transmission of said program.

16. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform an optimization method for improving the processing efficiency of a program, said method comprising the steps of:

when a program includes a branch, selecting at said branch a specific path from which to extract, relative to said branch, a series of paths that are not merged; and

performing optimization for said paths that are selected.